**CLAIMS**

Sub A1

1. A system for controlling traffic congestion within a buffered data switching network having a predetermined total buffer size, said system comprising:

(a) a packet counter for counting the number of newly arriving packets; and

(b) threshold means for setting a packet-count threshold;

wherein when the number of newly arriving packets reaches the packet-count threshold and when the average queue size exceeds the congestion threshold, a packet is discarded and the packet counter is reset to a zero count.

2. A system as in claim 1, further comprising calculation means for calculating an average queue size to be used by the threshold means in setting the packet-count threshold.

Sub A1

3. A system as in claim 2, wherein the calculation means regularly updates the average queue size using an exponential averaging technique.

4. A system as in claim 3, wherein the average queue size at time $t$ is calculated as:

$$\overline{Q}_t = \overline{Q}_{t-1} \times (1 - Alpha) + Q_t \times Alpha,$$

where $Q_t$ is an instantaneous queue size and $\overline{Q}_{t-1}$ is the average queue size at time $t$-1, and *Alpha* is a queue-length averaging parameter assigned a value between zero and one.

5. A system as in claim 4, wherein a progressively increasing value of *Alpha* is assigned with increasing level of traffic congestion.

6. A system as in claim 5, wherein the level of traffic congestion is indicated by the instantaneous queue size.

7. A system as in claim 3, wherein the average queue size is updated after a predetermined number of cells have arrived since a previous packet discard.

8. A system as in claim 3, wherein the average queue size is updated after a predetermined period of time has elapsed since a previous packet discard.

9. A system as in claim 2, further comprising means for dividing the total queue size into a pre-selected number of $N$ regions, wherein the threshold means sets the packet-count threshold by using a descending staircase function $F(n)$, such that one of every $F(n)$ packets is discarded, when the average queue size is in a buffer region $n$, $1 \leq n \leq N$.

10. A system as in claim 9, further comprising means for detecting traffic congestion by setting a congestion threshold and comparing the average queue size with the congestion threshold, such that a congestion condition is indicated by the average queue size being equal to or above the congestion threshold, and an absence of congestion is indicated otherwise.

11. A system as in claim 10, wherein the packet is discarded only during the congestion condition.

12. A system as in claim 10, wherein the packet counter begins to operate when traffic congestion is detected, and halts operation when an absence of traffic congestion is detected.

13. A system as in claim 2, further comprising means for dividing the total queue size into a pre-selected number of $M$ regions, for high-priority traffic defining a high-priority congestion threshold, and a pre-selected number of $N$ regions for low-priority traffic defining a low-priority congestion threshold, wherein the threshold means sets the packet-count threshold by using two functions $F(n,m)$ and $F(m)$, such that:

when the average queue size of high-priority traffic is above the high-priority congestion threshold and is in the buffer region $m$, $1 \leq m \leq M$, one of every $F(m)$ high priority packets is discarded; and

when the average queue size of low-priority traffic is above the low-priority congestion threshold and is in the buffer region $n$, $1 \leq n \leq N$, one of every $F(n,m)$ low priority packets is discarded.

14. A system as in claim 13, wherein the function $F(m)$ is a descending staircase function in the buffer region $m$, and the function $F(n,m)$, is a multivariable function of $m$ and $n$, which has a descending staircase behaviour in the buffer region $n$ for a fixed value of $m$.

15. A system as in claim 1, further comprising means for applying a priority scheme for discarding packets, which provides a differentiated service among service classes sharing a common buffer.

16. A system as in claim 1, wherein the threshold means uses a look-up table.

17. A system as in claim 1, wherein the threshold means sets the packet-count threshold upon arrival of a new packet into the system.

18. A system as in claim 1, wherein the threshold means sets the packet-count threshold upon departure of a packet from the system.

19. A method for controlling traffic congestion within a buffered data switching network having a predetermined total buffer size, said method comprising the steps of:

(a) counting the number of newly arriving packets;

(b) setting a packet-count threshold; and

(c) discarding a packet and resetting the packet-counter, when the number of newly arriving packets reaches the packet-count threshold and the average queue size exceeds the congestion threshold.

20. A method as in claim 19, further comprising a step of calculating an average queue size to be used by the step of setting the packet-count threshold.

Sub A7

21. A method as in claim 20, wherein the calculating step regularly updates the average queue size using an exponential averaging technique.

22. A method as in claim 21, wherein the average queue size at time $t$ is calculated as:

$$\overline{Q}_t = \overline{Q}_{t-1} \times (1 - Alpha) + Q_t \times Alpha,$$

where $Q_t$ is an instantaneous queue size and $\overline{Q}_{t-1}$ is the average queue size at time $t$-1, and *Alpha* is a queue-length averaging parameter assigned a value between zero and one.

23. A method as in claim 22, wherein a progressively increasing value of *Alpha* is assigned with increasing level of traffic congestion.

24. A method as in claim 23, wherein the level of traffic congestion is indicated by the instantaneous queue size.

25. A method as in claim 21, wherein the average queue size is updated after a predetermined number of cells have arrived since a previous packet discard.

26. A method as in claim 21, wherein the average queue size is updated after a predetermined period of time has elapsed since a previous packet discard.

27. A method as in claim 20, further comprising a step of dividing the total buffer size into a pre-selected number of $N$ regions, wherein the setting step sets the packet-count threshold by using a descending staircase function $F(n)$, such that one of every $F(n)$ packets is discarded, when the average queue size is in a buffer region $n$, $1 \leq n \leq N$.

28. A method as in claim 27, further comprising a step of detecting traffic congestion by setting a congestion threshold and comparing the average queue size with the congestion threshold, such that a congestion condition is indicated by the average queue size being above the congestion threshold, and an absence of congestion is indicated otherwise.

29. A method as in claim 28, wherein the packet is discarded only during the congestion condition.

30. A method as in claim 28, wherein the packet counter begins to operate when traffic congestion is detected, and halts operation when an absence of traffic congestion is detected.

31. A method as in claim 20, further comprising a step of dividing the total buffer size into a pre-selected number of $M$ regions, for high-priority traffic defining a high-priority congestion threshold, and a pre-selected number of $N$ regions for low-priority traffic defining a low-priority congestion threshold, wherein the setting step sets the packet-count threshold by using two functions $F(n,m)$ and $F(m)$, such that:

when the average queue size of high-priority traffic is above the high-priority congestion threshold and is in the buffer region $m$, $1 \leq m \leq M$, one of every $F(m)$ high priority packets is discarded; and

when the average queue size of low-priority traffic is above the low-priority congestion

threshold and is in the buffer region $n$, $1 \leq n \leq N$, one of every $F(n,m)$ low priority packets is discarded.

32. A method as in claim 31, wherein the function $F(m)$ is a descending staircase function in the buffer region $m$, and the function $F(n,m)$, is a multivariable function of $m$ and $n$, which has a descending staircase behaviour in the buffer region $n$ for a fixed value of $m$.

33. A method as in claim 19, further comprising a step of applying a priority scheme for discarding packets, which provides a differentiated service among service classes sharing a common buffer.